**Experiment 4**

# Rushikesh Ghige D15A Batch A Roll No: 21

**AIM: To create an interactive Form using form widget THEORY:**

**User Interface Design:** Interactive forms require a well-designed user interface (UI) to ensure

ease of use and clarity for users. Considerations include layout, labeling of fields, grouping related fields, and providing appropriate input controls (text fields, checkboxes, radio buttons, dropdown menus, etc.).

**Form Structure:** Define the structure of the form, including the types of information you want to collect from users. This could include personal details, contact information, preferences, or any other relevant data.

**Form Widgets:** Form widgets are the interactive elements used to collect data from users. Common form widgets include text fields, text areas, checkboxes, radio buttons, dropdown menus, and buttons. Each widget serves a specific purpose and is used to capture different types of input.

**Validation:** Implement validation rules to ensure that the data entered by users is correct and complete. Validation can include checking for required fields, enforcing data formats (e.g., email addresses, phone numbers), and validating input ranges or constraints.

**Error Handling:** Provide feedback to users when errors occur during form submission or validation. This helps users understand what went wrong and how to correct it. Error messages should be clear, concise, and displayed near the relevant input fields.

**Accessibility:** Ensure that the form is accessible to all users, including those with disabilities. This involves using semantic HTML, providing appropriate labels and instructions, ensuring keyboard navigation, and testing with assistive technologies.

**Submission Handling:** Define how form submissions are processed. This may involve sending data to a server-side script for processing, storing data in a database, or triggering other actions based on user input.

**User Experience (UX):** Consider the overall user experience when designing the form. Aim for simplicity, clarity, and efficiency to minimize user frustration and increase completion rates.

**Testing and Iteration:** Test the form extensively to identify any usability issues, bugs, or errors. Gather feedback from users and iterate on the design based on their input to improve the overall user experience.

# Code:

**main.dart : Main entry point of our flutter appimport 'package:firebase\_core/firebase\_core.dart';**

**import 'package:flutter/material.dart';**

**import 'package:google\_play\_clone/includes/myNavigationDrawer.dart';**

**import 'package:google\_play\_clone/includes/search\_bar\_module.dart';**

**import 'package:google\_play\_clone/pages/default\_for\_you.dart';**

**import 'package:google\_play\_clone/pages/game\_for\_you.dart';**

**import 'package:google\_play\_clone/pages/home\_for\_you.dart';**

**import 'package:google\_play\_clone/splash.dart';**

**import 'Authentication/auth.dart';**

**import 'firebase\_options.dart';**

**import 'home.dart';**

**void main() async{**

**WidgetsFlutterBinding.ensureInitialized();**

**await Firebase.initializeApp(**

**options: DefaultFirebaseOptions.currentPlatform,**

**);**

**runApp(MyApp());**

**}**

**class MyApp extends StatelessWidget {**

**// This widget is the root of your application.**

**@override**

**Widget build(BuildContext context) {**

**return MaterialApp(**

**title: 'Google Play Clone',**

**theme: ThemeData(**

**primaryColor: Color(0xff01865f),**

**),**

**home: LandingPage(),**

**debugShowCheckedModeBanner: false,**

**);**

**}**

**}**

**class MyDashboard extends StatefulWidget {**

**@override**

**\_MyDashboardState createState() => \_MyDashboardState();**

**}**

**class \_MyDashboardState extends State<MyDashboard> with SingleTickerProviderStateMixin {**

**late TabController \_googlePlayStoreController;**

**@override**

**void initState() {**

**super.initState();**

**\_googlePlayStoreController = TabController(vsync: this, initialIndex: 0, length: 5);**

**}**

**@override**

**Widget build(BuildContext context) {**

**return Scaffold(**

**appBar: AppBar(**

**title:Text("Google Play"),**

**actions: <Widget>[**

**IconButton(**

**icon: Icon(Icons.search),**

**onPressed: (){**

**showSearch( context: context, delegate:GooglePlaySearch() );**

**},**

**),**

**Icon(Icons.keyboard\_voice),**

**Padding(padding: EdgeInsets.only(right: 20.0)),**

**],**

**bottom: TabBar(**

**controller: \_googlePlayStoreController,**

**indicatorColor: Colors.white,**

**isScrollable: true,**

**tabs: <Widget>[**

**Tab(text: "HOME"),**

**Tab(text: "GAMES"),**

**Tab(text: "MOVIES"),**

**Tab(text: "BOOKS"),**

**Tab(text: "MUSIC"),**

**],**

**),**

**),**

**body: TabBarView(**

**controller: \_googlePlayStoreController,**

**children: <Widget>[**

**HomeForYou(),**

**GameForYou(),**

**DefaultForYou(),**

**DefaultForYou(),**

**DefaultForYou(),**

**],**

**),**

**drawer: Drawer(**

**child: MyNavigationDrawer(),**

**),**

**);**

**}import 'dart:convert';**

**import 'package:flutter/material.dart';**

**import 'package:intl/intl.dart';**

**import 'package:http/http.dart' as http;**

**import '../main.dart';**

**final formatter = DateFormat.yMd();**

**var enteredname='';**

**var entereddate;**

**class SignUp extends StatefulWidget {**

**const SignUp({super.key});**

**@override**

**State<StatefulWidget> createState() {**

**return \_SignUp();**

**}**

**}**

**class DT{**

**DT({required this.date});**

**final DateTime date;**

**String get formattefDate {**

**return formatter.format(date);**

**}**

**}**

**class \_SignUp extends State<SignUp> {**

**DateTime? \_selectedDate=DateTime.now();**

**void \_presentDatePicker() async {**

**final now = DateTime.now();**

**final firstDate = DateTime(now.year - 1, now.month, now.day);**

**// final lastDate = DateTime(now.year + 3, now.month, now.day);**

**final pickedDate = await showDatePicker(**

**context: context,**

**initialDate: now,**

**firstDate: firstDate,**

**lastDate: now);**

**if(\_selectedDate!=null){**

**setState(() {**

**\_selectedDate = pickedDate??\_selectedDate;**

**});}**

**}**

**String get formattedDate {**

**return formatter.format(DateTime.now());**

**}**

**void savedata()async{**

**enteredname=nameController.text;**

**final url=Uri.https('flutter-prep-5b74d-default-rtdb.firebaseio.com','user-data.json');**

**final response= await http.post(url,headers: {**

**'Content-type':'application/json'**

**},**

**body: json.encode({**

**'name': enteredname,**

**'dateofbirth': formatter.format(\_selectedDate!)**

**})**

**);**

**}**

**final \_dateController = TextEditingController();**

**final nameController=TextEditingController();**

**@override**

**Widget build(BuildContext context) {**

**DateTime? selectedDate=DateTime.now();**

**// var selectedDate = \_selectedDate;**

**return Scaffold(**

**backgroundColor: Theme.of(context).primaryColorLight,**

**body: Center(**

**child: Padding(**

**padding: const EdgeInsets.all(12.0),**

**child: Column(**

**mainAxisSize: MainAxisSize.min,**

**mainAxisAlignment: MainAxisAlignment.center,**

**crossAxisAlignment: CrossAxisAlignment.center,**

**children: [**

**TextField(**

**controller: nameController,**

**keyboardType: TextInputType.name,**

**decoration: InputDecoration(**

**border: OutlineInputBorder(**

**borderRadius: BorderRadius.all(Radius.circular(8))),**

**iconColor: Colors.black,**

**label: Text(**

**"Add Your Name",**

**style: TextStyle(**

**color: Colors.black,**

**fontStyle: FontStyle.italic,**

**),**

**),**

**),**

**expands: false,**

**),**

**const SizedBox(height: 10,),**

**const SizedBox(height: 10),**

**Row(**

**children: [**

**Expanded(**

**child: TextField(**

**controller: \_dateController,**

**decoration: InputDecoration(**

**label: Text('Enter your D.O.B',style: TextStyle(**

**color: Colors.black,**

**fontStyle: FontStyle.italic,**

**),**

**)**

**),**

**),**

**),**

**Expanded(**

**child: Row(**

**mainAxisAlignment: MainAxisAlignment.end,**

**crossAxisAlignment: CrossAxisAlignment.center,**

**children: [**

**Text(**

**selectedDate == null**

**? 'No Date Selected'**

**: formatter.format(\_selectedDate!),**

**style: const TextStyle(color: Colors.black)**

**),**

**const SizedBox(width: 10),**

**IconButton(**

**onPressed: () {**

**\_presentDatePicker();**

**},**

**icon: const Icon(**

**Icons.calendar\_month,**

**color: Colors.black,**

**)),**

**// const SizedBox(width: 16),**

**// Text(formattedDate),**

**],**

**),**

**),**

**],**

**),**

**const SizedBox(height: 10,),**

**ElevatedButton(onPressed: (){**

**savedata();**

**Navigator.of(context).push(MaterialPageRoute(builder: (ctx)=>**

**MyDashboard())**

**);**

**}, child: const Text("Next")),**

**],**

**),**

**),**

**),**

**);**

**}**

**}**

